# Third Review Presentation

## Cellular Automata and Algorithmic Preprocessing to Improve Clustering

### Guide

Kamalika Bhattacharjee

Dr. Kamalika Bhattacharjee - Computer Science and Engineering

### Team 8

G. Sneha

Vsubramanian

[illegible]

Ganta Sneha Rao - 106119037 Subramanian V V - 106119123 Viswonathan Manoranjan - 106119145

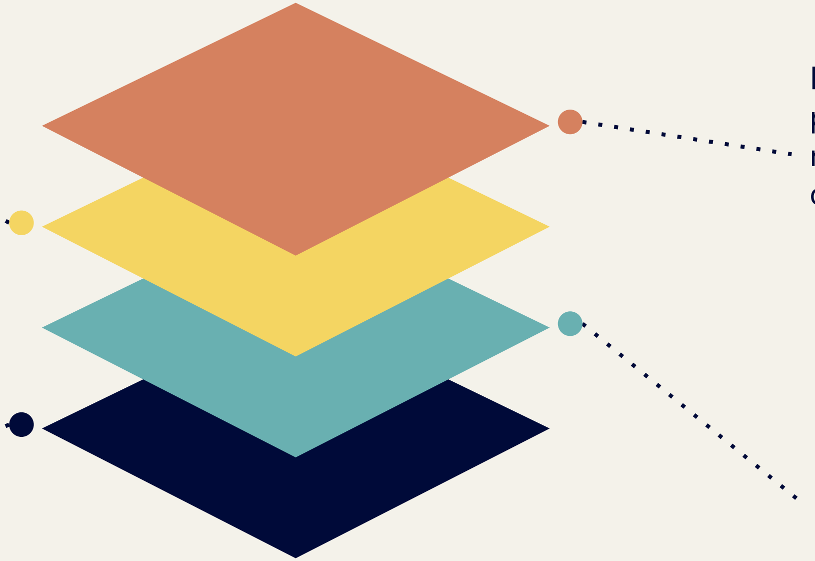
## Problem Statement

- With Reversible Cellular Automata acting as natural clusters, we explore the possibility of using reversible CA rules and taking a  non-conventional approach towards clustering numerical and categorical datasets, by grouping similar data points based on the rule and cycle formation using cellular automata.
- The goal is to propose a suitable encoding for the dataset, provide an efficient algorithm for clustering the data using the available CA rules and package our method for ease of access and use.

# Objectives

Determine a new encoding algorithm to avoid loss in data and improve the clustering silhouette score.

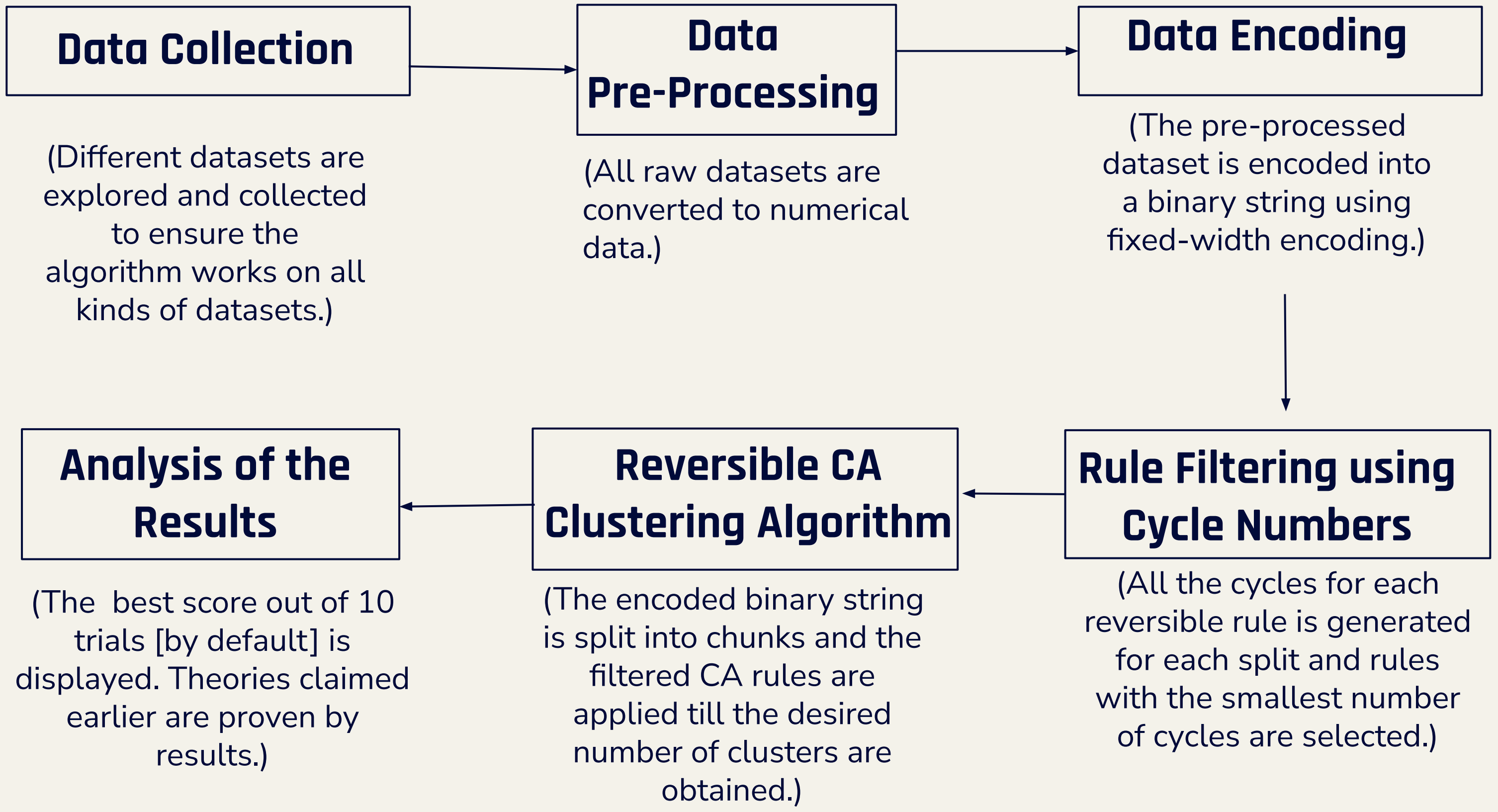
Package the whole process for ease of replication and development.



Propose a new algorithm that avoids the randomized search for rules for a particular dataset.

Provide flexibility for multiple datasets with varying number of records and features.

# Block Diagram



**Data Collection**

(Different datasets are explored and collected to ensure the algorithm works on all kinds of datasets.)

**Data Pre-Processing**

(All raw datasets are converted to numerical data.)

**Data Encoding**

(The pre-processed dataset is encoded into a binary string using fixed-width encoding.)

**Rule Filtering using Cycle Numbers**

(All the cycles for each reversible rule is generated for each split and rules with the smallest number of cycles are selected.)

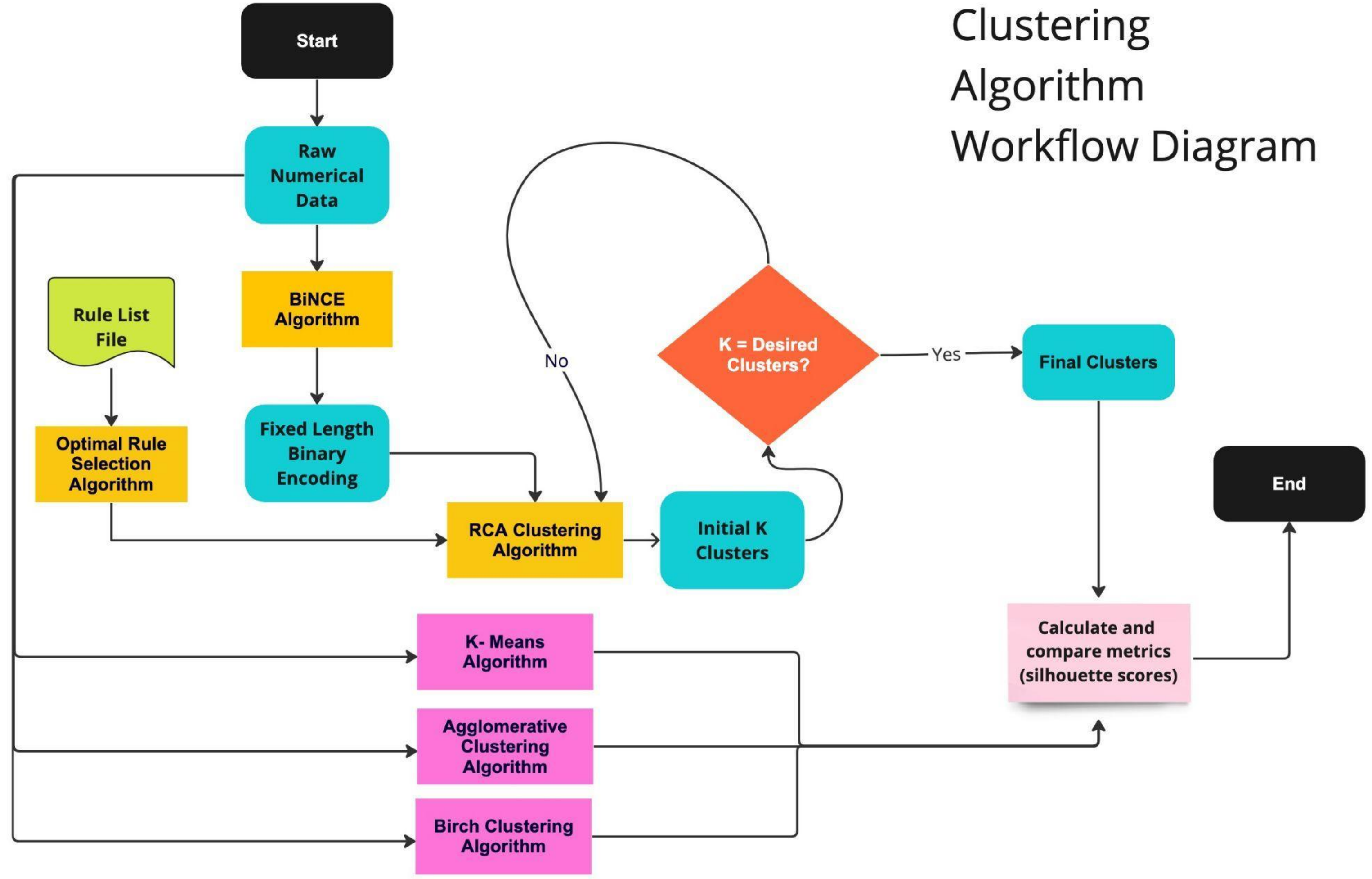
**Reversible CA Clustering Algorithm**

(The encoded binary string is split into chunks and the filtered CA rules are applied till the desired number of clusters are obtained.)

**Analysis of the Results**

(The  best score out of 10 trials [by default] is displayed. Theories claimed earlier are proven by results.)

# Clustering Algorithm Workflow Diagram


Start
Raw Numerical Data
BiNCE Algorithm
Rule List File
Fixed Length Binary Encoding
Optimal Rule Selection Algorithm
RCA Clustering Algorithm
Initial K Clusters
K = Desired Clusters?
No
Yes
Final Clusters
End
K- Means Algorithm
Agglomerative Clustering Algorithm
Birch Clustering Algorithm
Calculate and compare metrics (silhouette scores)
miro

# Algorithm of Modules

## Encoding Stage

We use our novel **Binary Normalised Ceiling Encoding (BiNCE)** algorithm, a fixed-width encoding technique to encode numerical and categorical datasets into binarized datasets.

**Step 1:** Normalize the dataset using Min-Max normalization to scale the score between 0 and 1

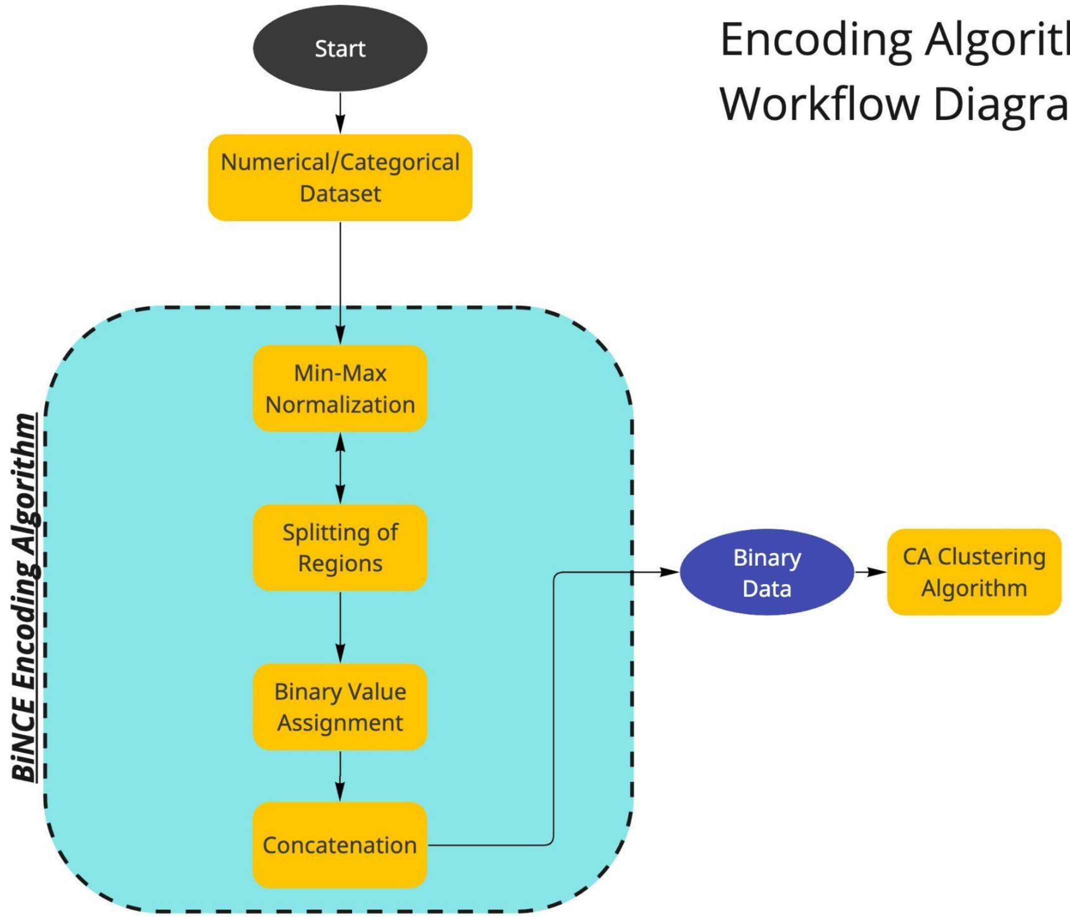
**Step 2:** Given the bit size of encoding for each feature, split the 0-1 region into the corresponding parts. For eg. if 2 bit encoding is used, the 0-1 region is then separated into $2^2 = 4$ regions i.e. 0-0.25, 0.25-0.5, 0.5-0.75 and 0.75-1. Similarly, for n-bit encoding, the 0-1 region is split into n regions.

**Step 3:** Each of these regions is assigned a n-bit binary code in ascending order of value. For eg.

|  |  |
|---|---|
| **0-0.25** | ***00*** |
| **0.25-0.5** | ***01*** |
| **0.5-0.75** | ***10*** |
| **0.75-1** | ***11*** |

**Step 4:** The features are then concatenated horizontally to get a bitstring representing one single data. Length of the bitstring will be ***n*l****, where* ***n*** *is the* ***bit size of each feature*** *in step 2, and* ***l*** *is the* ***number of features*** *of each data.*

# Encoding Algorithm Workflow Diagram



# Algorithm of Modules

## Pre-Clustering Stage

**Stage 1: Vertical Splitting:**
Here, each encoded binary string of size n is split into some divisions. Given the size of split ‘n’ i.e. 8-13*, our vertical splitting algorithm separates the binary string into ‘n’ size splits and pads the remaining elements in the last split.

**Stage 2: Rule Selection:**
For each of the splits i.e. 8-13, we exhaustively find the cycle lengths of all possible bit strings for all of the 162 candidate CA rules which are reversible for the given split size. From the rules, we choose the ones with the smallest number of cycles.

With our rule selection algorithm, we eliminate the random selection of reversible rules and eliminate rules that will not give us good results. This reduces our computation time exponentially as the total number of trials reduces from ${}^{162}C_2$ = 13041, to ${}^{20}C_2$ = 190, and in some cases even lower if initial clusters are less than desired clusters.

* Chosen experimentally to balance the trade-off between speed and accuracy

## Algorithm of Modules

### Clustering Stage

**Stage 3:  First Level Clustering:**
On applying these rules to each of the vertical split, we distribute the partitioned configurations into some preliminary cycles. The (partitioned) configurations under the same cycle form a unique cluster. Let the new length of configurations is *L*. Given our desired number of clusters, *L'*, as input, if $L<L'$, the algorithm skips that rule set and continues. Otherwise, we move on to Stage 4.

**Stage 4: Getting Desired Number Of Clusters:**
We obtain desired number of clusters by merging the clusters till desired number is reached. We iterate through the clusters to find one with the least number of elements (smallest cluster). The elements in the smallest cluster are extracted from that cluster and added to other clusters. The best fit for each element is determined by the clusters which give the highest performance index scores [silhouette score]. This procedure is followed till we reach desired number.

# Implementation/ Simulation Environment

## Environment

- ❖ Python == 3.11.0
- ❖ Scikit-learn == 1.2.0
- ❖ Pandas == 1.5.2
- ❖ Numpy == 1.23.5

- Processor : Intel i7 11700K 16M Cache, up to 5.00 GH
- Graphics Card : NVIDIA GeForce RTX A5000 24GB GDDR6
- Disk : 1TB SSD

## Datasets

1. School District Breakdown Dataset
2. Iris Dataset
3. Customer Credit Card Information Dataset

01

### Python

For ease of understanding, functional programming, GPU compatibility and replicability of previous papers

02

### Numpy and Pandas

For data handling and faster/parallel computations.

03

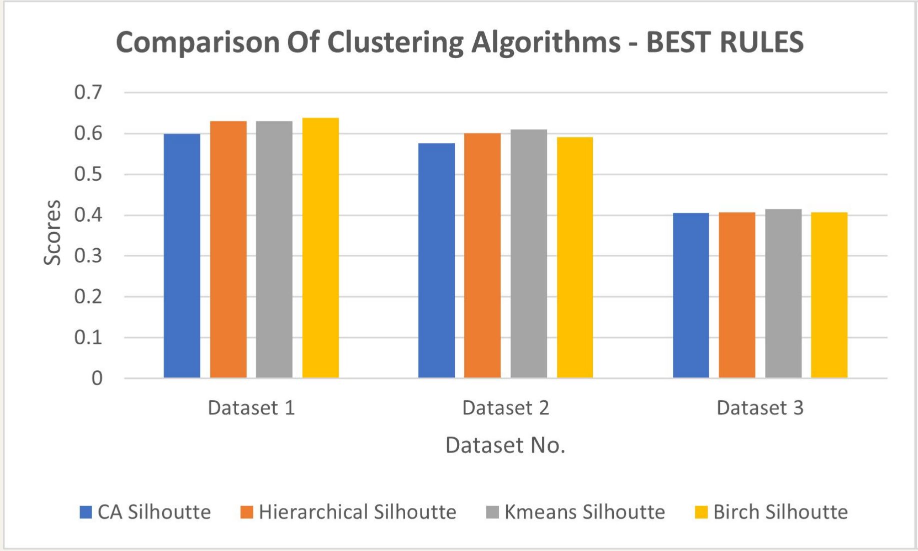
### Sklearn

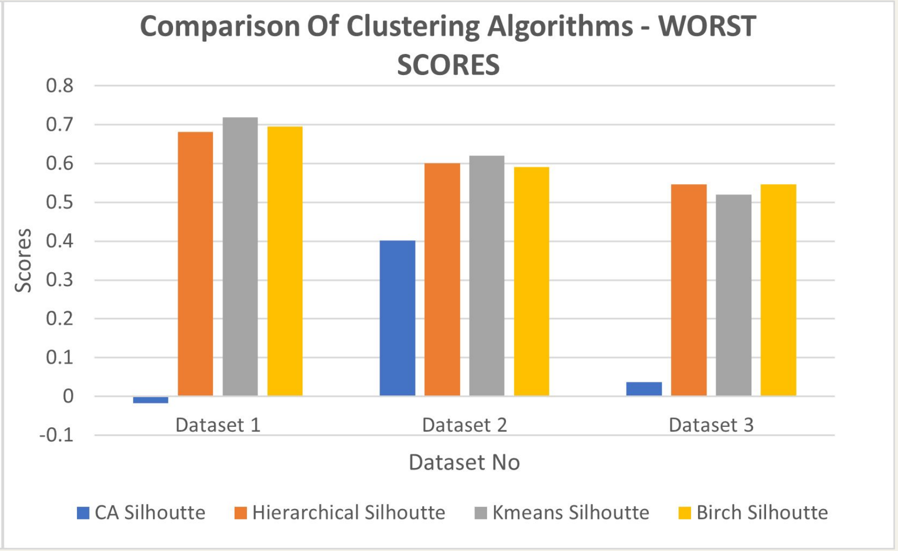
Employing K-Means, Birch and Agglomerative clustering for baseline scores and calculating silhouette scores

## Results

**Comparison of our CA clustering algorithm with K-Means, Hierarchical and Birch clustering algorithms to give a better measure of its performance.**

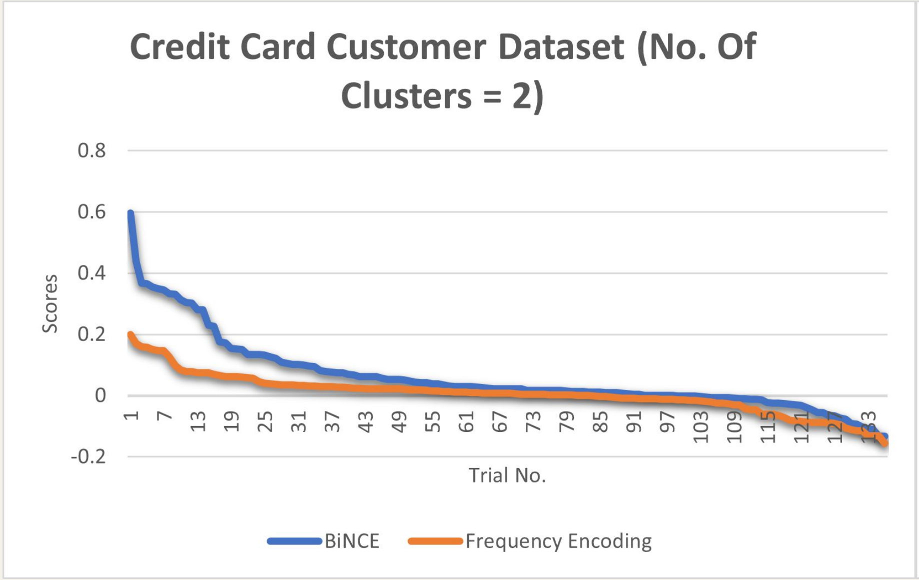
The graph below shows that the rules selected in order to reduce the search space return the best scores and are comparable to that of the other clustering algorithms. Similarly, we can also see that the rules that are not included in the search space are clearly returning low scores and are not comparable to the other clustering algorithms.

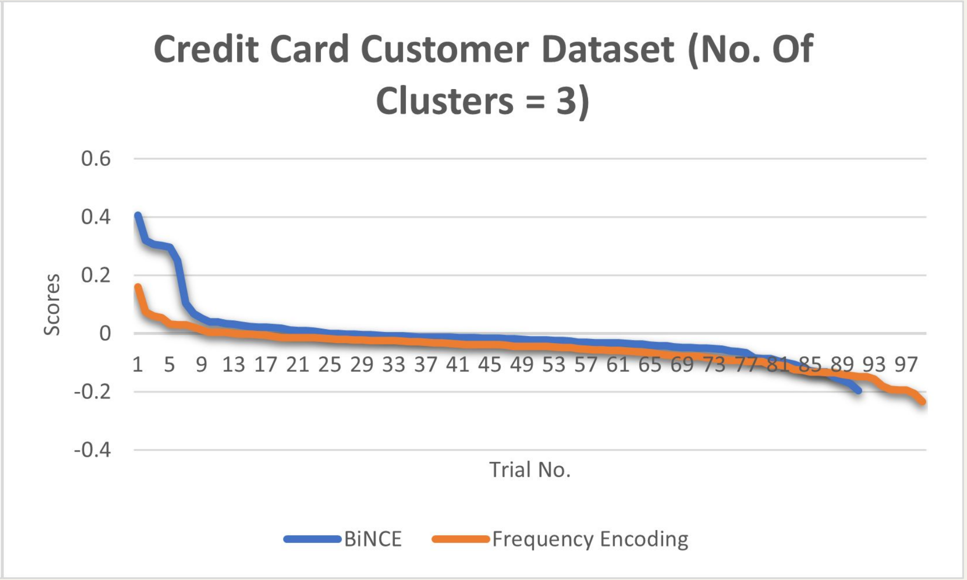




## Results

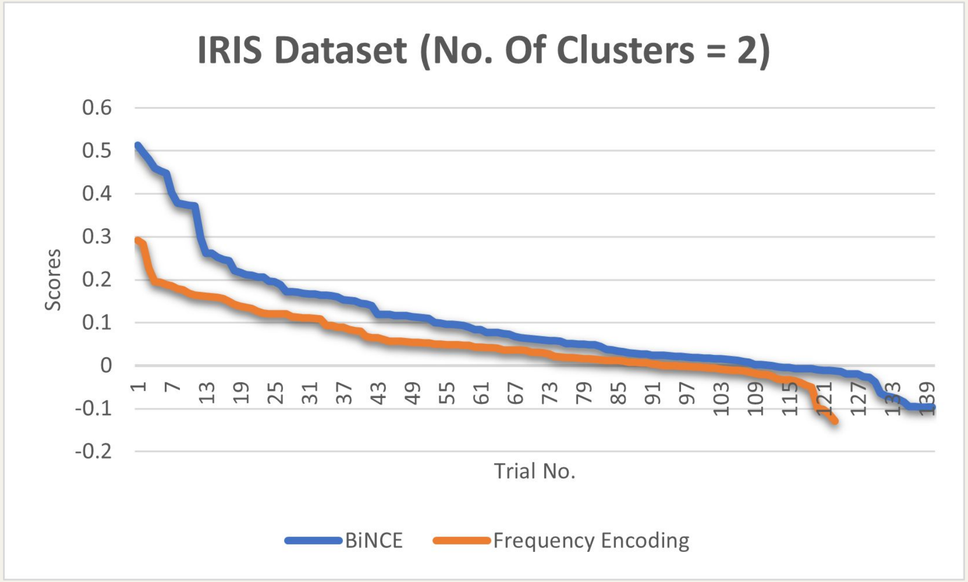
Compared to the previously used **Frequency Based Encoding**, the novel **BiNCE** encoding algorithm results in less loss of data during encoding numerical i.e. floating point and decimal datasets, hence leading to competitive scores with that of state-of-the-art clustering methods.





## Results

Compared to previously used **Frequency Based Encoding**, the novel **BiNCE** encoding algorithm results in less loss of data during encoding numerical i.e. floating point and decimal datasets, hence leading to competitive scores with that of state-of-the-art clustering methods.



# References

[1] *Sukanya Mukherjee, Kamalika Bhattacharjee, and Sukanta Das. 2020.* **Cycle Based Clustering Using Reversible Cellular Automata.** *In Cellular Automata and Discrete Complex Systems: 26th IFIP WG 1.5 International Workshop, AUTOMATA 2020, Stockholm, Sweden, August 10–12, 2020, Proceedings. Springer-Verlag, Berlin, Heidelberg, 29–42.*

[2] *S. Mukherjee, K. Bhattacharjee and S. Das*, "**Clustering Using Cyclic Spaces of Reversible Cellular Automata**," *Complex Systems, 30(2), 2021 pp. 205–237.*

[3] *Hong He, Yonghong Tan*, "**Automatic pattern recognition of ECG signals using entropy-based adaptive dimensionality reduction and clustering**", *Applied Soft Computing, Volume 55, 2017*

[4] *Mukherjee, Sukanya & Bhattacharjee, Kamalika & Das, Sukanta. (2021).* "**Reversible Cellular Automata: A Natural Clustering Technique. Journal of Cellular Automata**". *16. 1-38.*

[5] *Abhishek S, Mohammed Dharwish, Amit Das, and Kamalika Bhattacharjee*, "**A Cellular Automata Based Clustering Technique for High-Dimensional Data**", *Proceedings of the Second Asian Symposium on Cellular Automata Technology 2023, IIEST Shibpur*